

Estimación

Estimación: Es una predicción de lo que puede llegar a pasar

A medida que el proyecto avance la incertidumbre disminuye

Por eso es importante hacerla al principio, como la incertidumbre va disminuyendo la estimación debe ir actualizándose conforme vamos ganando experiencia.

Si hablamos de estimación tradicional normalmente decimos que el orden en que se estima es, primero el tamaño del producto a construir, segundo el esfuerzo(horas lineales), luego cuanto tiempo calendario y finalmente el costo que me llevara el producto.

Cuando hablamos de Agile esto cambia.

Estimación es diferente a planificación, la estimación es la entrada que voy a tener para hacer mi proceso de planificación.

La estimación es una técnica que uso para poder predecir el tamaño, complejidad y esfuerzo que me llevara del producto...

La estimación en sí mismo no es un compromiso si no un respaldo sobre el proceso de planificación, me sirve para que el proceso de planificación tenga cierta credibilidad.

Tips respecto de las estimaciones...

- ⚠ Si las estimaciones se utilizan como compromisos son muy peligrosas y perjudiciales para cualquier organización.
- 📅 Lo más beneficioso en las estimaciones es el "proceso de hacerlas".
- ✓ La estimación podría servir como una gran respuesta temprana sobre si el trabajo planificado es factible o no.
- ✗ La estimación puede servir como una gran protección para el equipo.

Estimaciones Ágiles

- Las features/stories son estimadas usando una medida de tamaño relativo conocida como story points (SP)
- Las medidas relativas no son absolutas.
- Story Points no es una medida basada en tiempo



Las estimaciones en general se van a circunscribir en una primera instancia por hacer una estimación por cada una las user stories.

La unidad de medida de una estimación ágil son los Story Points, las medidas en el caso de las estimación ágiles son medidas relativas en comparación a la tradicionales(por ejem horas lineales)

En agile las estimaciones surgen por una comparación, esa medida de SP tendrá un valor diferente dependiendo del equipo que haga la estimación.

Estas medidas involucra 3 dimensiones, la complejidad de la US, el esfuerzo que me tomara construir la US y la incertidumbre con la que estoy haciendo la estimación.

Estas 3 me determinaran una cantidad determinada de SP.

Estimación Relativa

- Las personas no saben estimar en términos absolutos
- Somos buenos comparando cosas
- Comparar es generalmente más rápido.
- Se obtiene una mejor dinámica grupal y pensamiento de equipo más que individual
- Se emplea mejor el tiempo de análisis de las stories

800
700
600
500
400
300
200
100

y...pero, el tamaño?

- “La palabra tamaño refiere a cuan grande o pequeño es algo”
- El tamaño es una medida de la cantidad de trabajo necesaria para producir una feature/story.
- El tamaño indica:
 - Cuán compleja es una feature/story
 - Cuánto trabajo es requerido para hacer o completar una feature/story
 - Y cuán grande es una feature/story



El tamaño y es esfuerzo son diferentes en este contexto... el esfuerzo tiene que ver con cuanto voy a necesitar yo para poder implementar la US (por ejem: esa US me llevara X horas)... y el tamaño en agile nos referimos a la estimación realizada en términos de SP.. (complejidad, esfuerzo e incertidumbre)

Tamaño Vs. Esfuerzo



Las estimaciones basadas en tiempo son más propensas a errores debido a varias razones.

- Habilidades
- Conocimiento
- Asunciones
- Experiencia
- Familiaridad con los dominios de aplicación/negocio



Tamaño **NO ES** esfuerzo

Le podemos asignar a una US diferentes tamaños relativos (nosotros trabajaremos con la serie de Fibonacci)

Y qué hacemos con el tamaño!?????

- Tamaño por números: 1 a 10
- Talles de remeras: S, M, L, XL, XXL
- Serie 2ⁿ : 1, 2, 4, 8, 16, 32, 64, etc.
- Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, etc.

Una vez elegida la escala no se cambia! Si se cambia cambiamos el metro patrón

Story Point:

Es una unidad de medida específica (del equipo) de, complejidad, riesgo y esfuerzo, es lo que “el kilo” a la unidad de nuestro sistema de medición de peso

Story point da idea del “peso” de cada story y decide cuan grande (compleja) es

La complejidad de una

- feature/story tiende a
- incrementarse exponencialmente

Me sirve para determinar el peso que tendrá una US cuando tenga que implementarla.

Como la complejidad de la US tiende a incrementarse exponencialmente, la serie de Fibonacci es una buena forma de medir o darle valores relativos a esa manera exponencial e incrementarse.

ACA MEDIMOS LAS US EN TERMINOS DE TALLES DE REMERAS

3 Stories que queremos estimar

Story 1



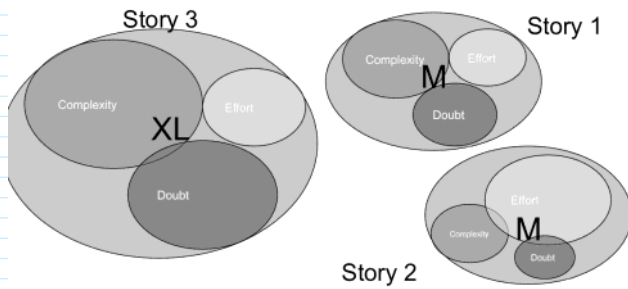
Story 2



Story 3



“tamaño” de las Stories



Velocidad

Velocidad/ Velocity es una medida (métrica) del progreso de un equipo. Se calcula sumando el número de story points (asignados a cada user story) que el equipo completa durante la iteración.

Se cuentan los story points de las Users Storys que están completas, no parcialmente completas.

La Velocidad corrige los errores de estimación

Una técnica de estimación que sumamos es Poker Planning ... no es propia de Agile pero la usamos igual.

¿Cómo “decodificar” las estimaciones?



- **0:** Quizás ud. no tenga idea de su producto o funcionalidad en este punto.
- **1/2, 1:** funcionalidad pequeña (usualmente cosmética).
- **2-3:** funcionalidad pequeña a mediana. Es lo que queremos. 😊
- **5:** Funcionalidad media. Es lo que queremos 😊
- **8:** Funcionalidad grande, de todas formas lo podemos hacer, pero hay que preguntarse sino se puede partir o dividir en algo más pequeño. No es lo mejor, pero todavía 😊
- **13:** Alguien puede explicar por que no lo podemos dividir?
- **20:** Cuál es la razón de negocio que justifica semejante story y más fuerte aún, por qué no se puede dividir?.
- **40:** no hay forma de hacer esto en un sprint.
- **100:** confirmación de que está algo muy mal. Mejor ni arrancar.

Poker Planning

Prerrequisitos:

- Lista de features/stories a ser estimadas
- Cada estimador tiene un mazo de cartas.

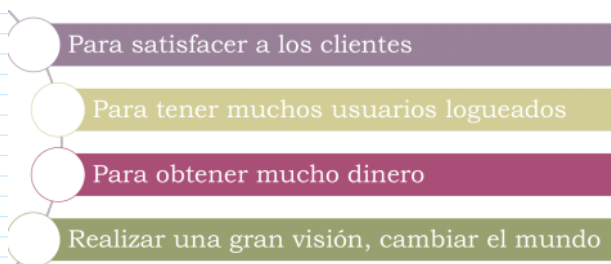
Pasos:

1. Determine la **base story** (la canónica) que será usada para comprar con las otras stories. Digamos, Story Z.
 - 1.1 La story a ser estimada se lee a todo el equipo.
 - 1.2 Los estimadores discuten la story, haciendo preguntas al product owner (las que se necesiten).
 - 1.3 Cada estimador selecciona una carta y pone la carta boca abajo en la mesa.
 - 1.4 Cuando todos pusieron las cartas, las mismas se exponen al mismo tiempo.
 - 1.5 Si todos los estimadores selecciona el mismo valor, ese es el estimado. Sino, los estimadores discuten sus resultados, poniendo especial atención en los más altos y los más bajos. Después de la charla, GOTO to 1.3.
2. Se toma la próxima story, se discute con el product owner.
3. Cada estimador asigna a la story un valor por comparación contra la base story. “Cuan grande/pequeña, compleja, riesgosa es esta story comparada con Story Z?”. GOTO to 1.3

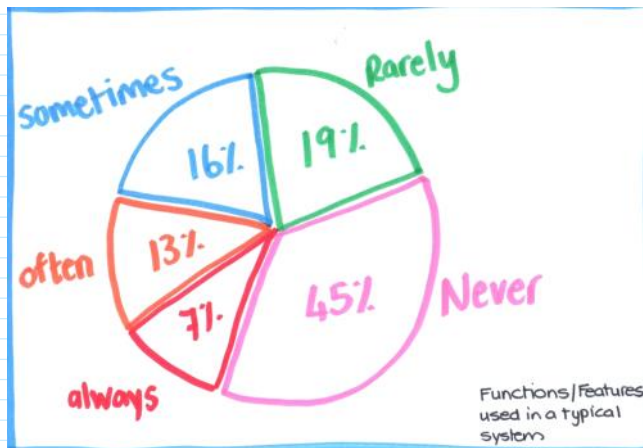


Gestión de Productos

¿Por qué creamos productos?



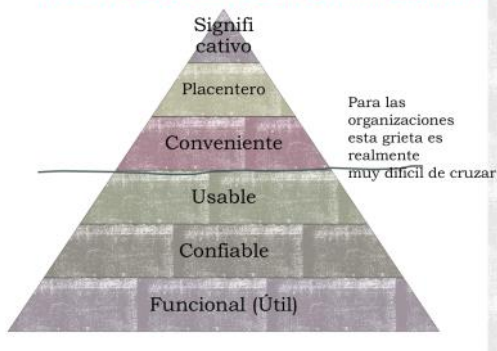
¿Qué características realmente utilizamos de un producto de software?



La mayoría de los productos de SW exitosos tienen solamente un grupo de funcionalidades que se usan a menudo.. Y el resto es plata que gastamos y para funcionalidades que no usamos.

Evolución de los Productos de Software

Focalizado en experiencias (gente, actividades, contexto)

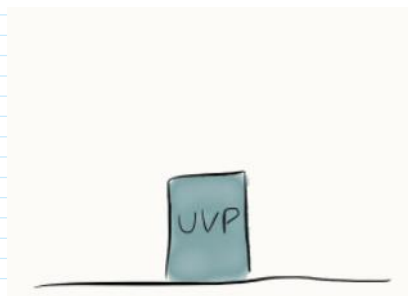


El objetivo del MVP es crear un mínimo producto que me permita evaluar si el producto que voy a construir tiene público. Necesito plantear una hipótesis y validarla... el MVP me permite comprobar la hipótesis en caso que no se cumpla construyo otro MVP y así... si la hipótesis resulta falsa y no use MVP termine gastando mucho por eso usamos MVPs

Explicando MVPs, MVFs, MMFs con el Dinosaurio Lean/Agile

- Comprender que un producto nuevo tiene una hipótesis de valor único: Producto / servicio será único.

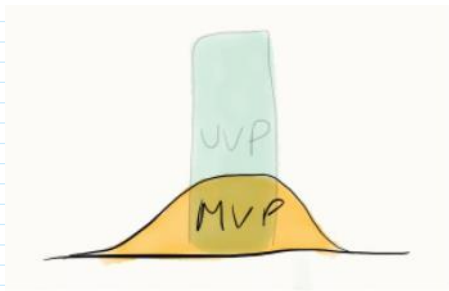
Un producto tiene un valor único, yo espero que este producto me sirva para



- El siguiente paso es crear un producto mínimo viable (MVP) para probar su hipótesis.

El **MVP** será una partecita del UVP, conjunto de funcionalidades mínimos que me permitan validar mi hipótesis de que ese valor único es algo interesante para el usuario..

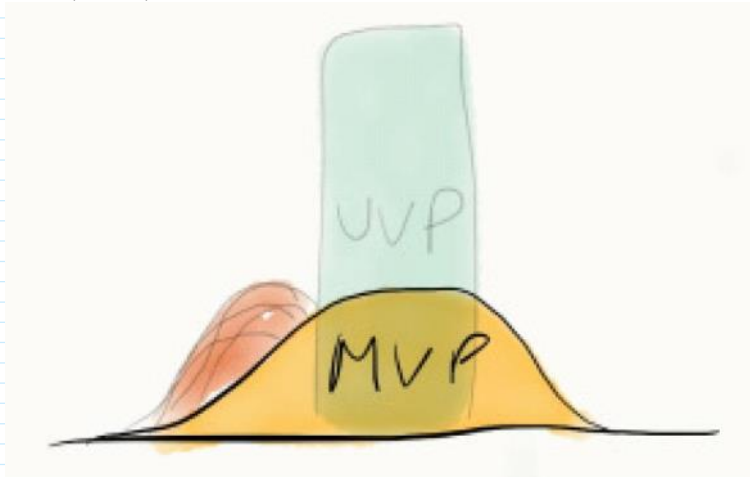
- Esto se centra en su propuesta de valor única, pero normalmente también proporciona un poco de funciones de "Table stakes" sólo para asegurarse de que sea "viable" como producto.



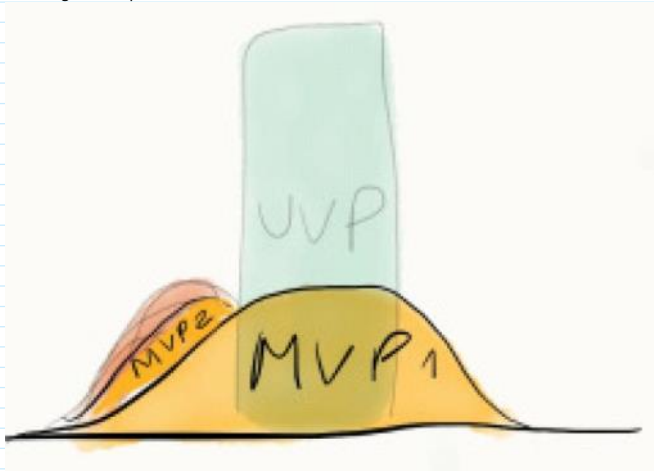
- Tu **MVP** también es una hipótesis. Podría ser lo suficientemente bueno para encontrar un mercado o no.
- Se muestra el caso en el que cada cliente potencial con el que interactúas te dice "Esto es genial, pero para poder usarlo necesito X" y X es diferente para cada cliente / usuario.
- Esto muestra que aún no se encuentra un mercado para el producto



- Si por el contrario ves cada vez más respuestas apuntando al MISMO X entonces tiene sentido revisar la hipótesis de Cliente/Problema/Solución

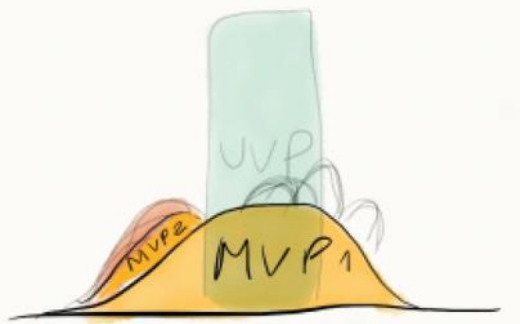


- Básicamente, estás ejecutando un pivot.
- Está construyendo **MVP2** centrado en la nueva hipótesis basada en el aprendizaje reciente de Desarrollo de clientes generado por el anterior **MVP**

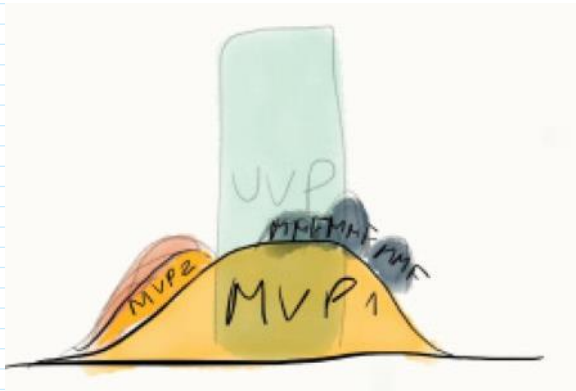


- Supongamos que **MVP2** es exitoso y está viendo una tracción real de los primeros usuarios.
- Si desea aumentar el crecimiento y busca una penetración más profunda de sus primeros usuarios además de atraer nuevos clientes, algunos de ellos más allá de la multitud de usuarios pioneros.
- En función del feedback recopilado y la investigación de su gestión de productos, tiene un par de áreas que potencialmente pueden traer este crecimiento.

- Algunos de ellos por cierto amplían su propuesta de valor única y algunos hacen que tu producto actual sea más robusto



- En el caso de áreas con una fuerte indicación de valor, puede directamente definir un **MMF** (Características mínimas comercializables). Para encontrar la pieza mínima que pueda empezar a traer crecimiento.
- El objetivo del **MMF** es aportar valor. Supone una alta certeza de que existe valor en esta área y que sabemos cuál debe ser el producto para proporcionar este valor.
- La razón para dividir una característica grande en **MMF** más pequeños es principalmente el tiempo de comercialización (Time to market) y la capacidad de aportar valor en muchas áreas.
- Una indicación de que está trabajando en MMF es que cuando al liberarse uno se siente cómodo trabajando en el próximo MMF en esa área.



- Si se desea esperar para ser si el primer MMF apesta ... entonces está de vuelta en la tierra de la hipótesis.
- Ahora tu hipótesis se centra en una característica en lugar del producto.
- Tienes un área con alto potencial pero también alta incertidumbre.

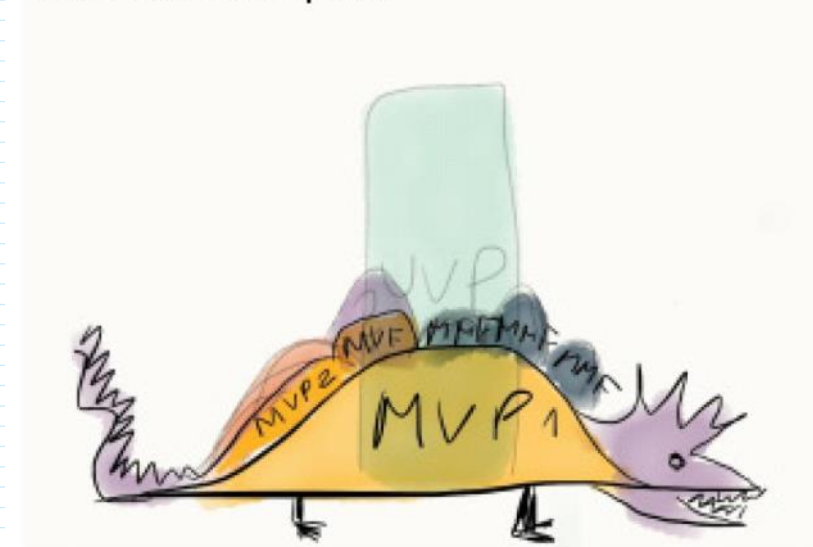


- La forma de afrontarlo es crear una función "pionera": MVF (Característica Mínima Viable). La característica mínima que aún puede ser viable para uso real y aprendizaje de los usuarios reales.
- Si la MVF resulta exitosa (hit gold), puede desarrollar más MMF en esa área para tomar ventaja (si eso tiene sentido). Si no es así, puede cambiar a otro enfoque hacia esa área de características, o en algún momento buscar una ruta de crecimiento alternativa. Esencialmente, el MVF es una versión mini del MVP.



- El producto se cultiva en mercados inciertos al intentar varios MVP.
- Cuando se logra ajustar el producto en el mercado de productos se combinan MMF y MVF según el nivel de incertidumbre del negocio / requisitos en las áreas en las que se está enfocando.
- Si bien los MVP / MMF / MVF son atómicos desde una perspectiva empresarial (no puede implementar y aprender de algo más pequeño) pueden ser bastante grandes desde la perspectiva de la implementación.
- El dinosaurio carpaccio se obtiene cortando cada una de esas piezas en pequeñas porciones destinadas a reducir el riesgo de ejecución / tecnología (normalmente se denominan User Stories).
- Esas porciones más pequeñas pueden tener un valor comercial tangible o no.
- Para otros puede ser importante proporcionar una temprana retroalimentación de decisiones a lo largo del camino

El modelo completo



Productos Mínimos para la Gestión de Productos...

Minimal Viable Product (Producto Mínimo Viable) : **(MVP)**

Minimal Viable Feature (Característica Mínima Viable) :Lo mismo del MVP pero en terminos de una features en vez de pensar en terrino de producto pienso en termino de característica. **(MVF)**

Minimal Release Feature (Características Mínimas del Release) : características mínimas para sacarla al mercado. **(MRF)**

Minimal Marketable Feature (Característica Mínima Comercializable): no se encarga de validación de hipótesis si no que característica debe tener mi producto para poder venderlo... apunto a comercializar. **(MMF)**



Relación entre MVP, MMF, MMP, MMR

MMP (Minimum Marketable Product):

Es el producto con la cantidad mínima de características para ser **comercializable y atractivo** para los usuarios en el mercado.

MRF (Minimal Release Feature)

- **Definición:** Se refiere a una **característica mínima que puede ser lanzada** en una nueva versión del producto. La MRF es una funcionalidad o conjunto pequeño de funcionalidades que pueden ser lanzadas de forma individual para aportar valor al usuario

MMF (Minimum Marketable Feature)

Similar al MRF, pero centrado en una característica mínima comercializable, es decir, una funcionalidad que tiene suficiente valor como para ser vendida o publicitada.

MMR (Minimum Marketable Release)

Relación entre MVP, MMF, MMP, MMR

Tengo un MVP, dividido a este en varios RELEASES, dentro de estos releases cada release incluye distintas características mínimas comercializables

Similar al MRF, pero centrado en una característica mínima comercializable, es decir, una funcionalidad que tiene suficiente valor como para ser vendida o publicitada.

MMR (Minimum Marketable Release)

- Definición: Es una **versión mínima del producto** que contiene suficientes funcionalidades nuevas y mejoras como para ser comercializada y tener un impacto relevante en el mercado.

Nivel de entrega: El MRF es una funcionalidad individual mínima, mientras que el MMR es una **versión** del producto que puede incluir múltiples funcionalidades (como varios MRFs).

RELACIÓN ENTRE MVP, MMF, MMP, MMR

MVP

- Versión de un **nuevo producto** creado con el **menor esfuerzo posible**
- Dirigido a un **subconjunto de clientes potenciales**
- Utilizado para obtener **aprendizaje validado**.
- Más cercano a los **prototipos que a una versión real funcionando de un producto**.

MMF

- es la **pieza más pequeña de funcionalidad** que puede ser liberada
- tiene valor tanto para la organización como para los usuarios.
- Es parte de un MMR or MMP.

MMP

- Primer release de un MMR dirigido a **primeros usuarios** (early adopters),
- Focalizado en características clave que satisfarán a este grupo clave.

MMR

- Release de un producto que tiene el conjunto de características más pequeño posible.
- El incremento más pequeño que ofrece un valor nuevo a los usuarios y satisface sus necesidades actuales.
- MMP = MMR1**

17

MVP (Minimal Viable Product - Producto Mínimo Viable)



Es un concepto de *Lean Startup* que enfatiza el impacto del aprendizaje en el desarrollo de nuevos productos.



Eric Ries: "versión de un nuevo producto que permite a un equipo recopilar la cantidad máxima de aprendizaje validado sobre clientes con el menor esfuerzo". Este aprendizaje validado viene en forma de si sus clientes realmente comprarán su producto.



Una premisa clave detrás de la idea de **MVP** es que usted produce un producto real que puede ofrecer a los clientes y observar su comportamiento real con el producto o servicio.



Ver lo que la gente realmente hace con respecto a un producto es mucho más confiable que preguntarle a la gente qué harían.

MVP: Minimal Viable Product



TIENE EL VALOR SUFICIENTE PARA QUE LAS PERSONAS ESTÉN DISPUESTAS A USARLO O COMPRARLO INICIALMENTE.



DEMUESTRA SUFICIENTE BENEFICIO FUTURO PARA RETENER A LOS PRIMEROS USUARIOS.



PROPORCIONA UN CICLO DE RETROALIMENTACIÓN PARA GUIAR EL DESARROLLO FUTURO.

MVF es una versión mini del MVP.

Característica a pequeña escala que se puede construir e implementar rápidamente, utilizando recursos mínimos, para una población objetivo para probar la utilidad y adopción de la característica.

Un MVF debe proporcionar un valor claro a los usuarios y ser fácil de usar.

MVF requiere recursos mínimos, los estándares de calidad de la industria y la producción deben guiar el diseño y la confiabilidad.

El grupo de usuarios para un MVF, son los primeros en adoptar, los clientes leales que han compartido conocimientos anteriormente o los miembros de una junta asesora de clientes.. Usuarios flexibles y tolerantes.

Los resultados le ayudarán a tomar decisiones estratégicas sobre productos

MVP vs MMF o MMP: Errores comunes



Confundir a un MVP, **que se enfoca en el aprendizaje**, con Característica Comercializable Mínima (MMF) o con Producto Comercializable Mínimo (MMP), ambos se enfocan en "ganar".



El riesgo de esto es entregar algo sin considerar si es lo correcto que satisface las necesidades del cliente.



Enfatizar la parte **mínima** de MVP con exclusión de la parte **viable**. El producto entregado no es de calidad suficiente para proporcionar una evaluación precisa de si los clientes utilizarán el producto.



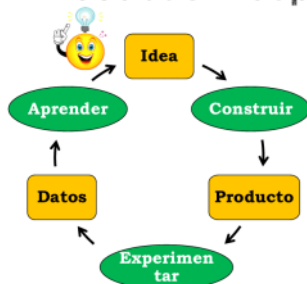
Entregar lo que consideran un MVP, y luego no hacer más cambios a ese producto, independientemente de los comentarios que reciban al respecto.

Valor vs. Desperdicio

- **Preguntas:** ¿Cuáles de nuestros esfuerzos crean valor y cuáles son desperdicio?
- *Lean Thinking* define la creación de valor como proveer beneficios a los clientes, cualquier otra cosa es desperdicio!
- La productividad de un Startup **no puede medirse en términos de cuánto se construye cada día**, por el contrario **se debe medir en términos de averiguar la cosa correcta a construir cada día**

Método Lean Start- up

Build-Experiment-Learn Feedback Loop



- El éxito no es entregar un producto, el éxito se trata de entregar un producto (o característica de producto) que el cliente usará.
- La forma de hacerlo es alinear los esfuerzos continuamente hacia las necesidades reales de los clientes.
- The **Build-Experiment-Learn feedback loop** permite descubrir las necesidades del cliente y alinearlas metodológicamente.

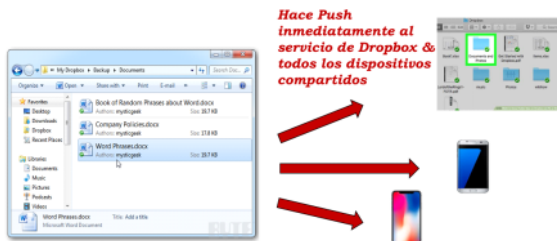
La fase construir: MVP

- Ingresar lo más rápido posible con un Producto mínimo viable (MVP).
- Un MVP varía en complejidad desde pruebas de humo (smoke tests) extremadamente simples (poco más que un anuncio) hasta prototipos tempranos



Ejemplo de MVP: Dropbox

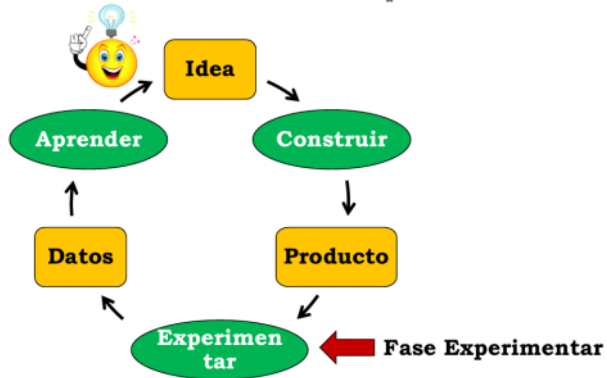
- Dropbox es una herramienta fácil de usar para compartir y sincronizar archivos, que utiliza una técnica de almacenamiento en caché de inserción (o replicación completa)



La fase construir: MVP

- Decidir qué tan complejo no se puede hacer un MVP de manera formulada:
 - ¡Requiere juicio!
 - En caso de duda, simplifique
 - Evite la construcción excesiva y la promesa excesiva
 - Cualquier trabajo adicional más allá de lo que necesita para comenzar el ciclo podría ser un desperdicio
- Un MVP no sólo habla sobre el diseño del producto y las preguntas técnicas, sino que también sirve para probar hipótesis comerciales fundamentales
- Por lo tanto, sirve para proporcionar una dosis necesaria de realidad

Build-Experiment-Learn Feedback Loop



Dilema: La audacia de Cero

- A menudo es más fácil recaudar dinero cuando tiene:
 - cero ingresos
 - cero clientes
 - tracción cero
 que cuando tienes una pequeña cantidad de cada uno!
- Cero invita a la imaginación, pero los números pequeños hacen preguntas sobre si los números grandes alguna vez se materializarán
- Este fenómeno (llamado **la audacia de cero**) crea un incentivo brutal:
- Aplazar el lanzamiento de cualquier versión de un producto hasta que esté seguro del éxito

Dilema: La audacia de Cero

- Si pospone experimentar con su MVP, surgirán algunos resultados desafortunados como:
 - La cantidad de trabajo desperdiciado puede aumentar
 - Se perderán los comentarios esenciales
 - El riesgo de que su startup construya algo que nadie quiere puede aumentar
- Compensaciones:
 - ¿Preferiría atraer capital de riesgo y potencialmente derrocharlo?
 - ¿O preferiría atraer capital de riesgo y utilizarlo sabiamente?
- Use un MVP para experimentar (inicialmente, en silencio) con los primeros usuarios en el mercado
- Verifique su concepto probando TODOS sus elementos, comenzando por los más riesgosos.



Los elementos más riesgosos del plan / concepto de una startup (es decir, las partes de las que todo depende) se denominan supuestos de salto de fe.



Por ejemplo, ¿Cuál fue el principal supuesto de salto de fe de Dropbox?



La sincronización de archivos es un problema



La mayoría de las personas no conocen una determinada solución (o incluso un problema); pero una vez que experimentan la solución, ¡no pueden imaginar cómo vivirían sin ella!

Supuestos de “Saltos de Fe”

Supuestos de “Saltos de Fe”



Hipótesis del valor:

- Prueba si el producto realmente está entregando valor a los clientes después de que comienzan a usarlo
- Una métrica de prueba: tasa de retención



Hipótesis de crecimiento:

- Prueba cómo nuevos clientes descubrirán el producto
- Una métrica de prueba: tasa de referencia o Net Promoter Score (NPS)

Ejemplo: Facebook

facebook

- En 2004, Facebook tenía 150,000 usuarios registrados con muy pocos ingresos.
- Sin embargo, ese verano recaudaron sus primeros \$ 500,000 en capital de riesgo
- Menos de un año después, recaudaron \$ 12.7 millones adicionales
- ¿Cómo Facebook pudo recaudar tanto dinero cuando su uso real era pequeño?

▪ Hipótesis de valor validada:

- Más de la mitad de los usuarios volvieron al sitio todos los días.

facebook

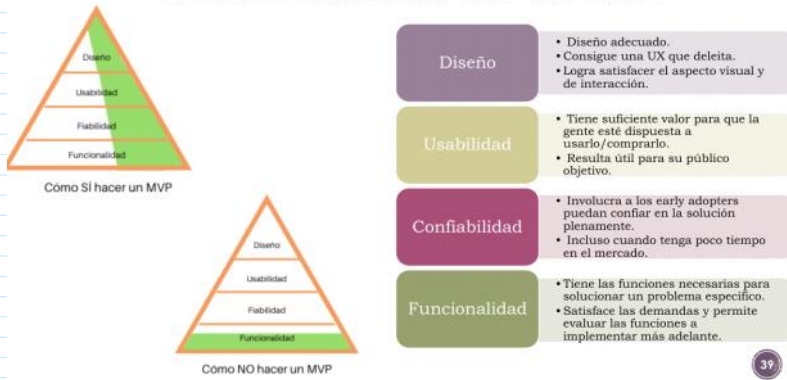
▪ Hipótesis de crecimiento validada:

- Facebook se lanzó el 4 de febrero de 2004, y para fines de ese mes, casi $\frac{3}{4}$ de los estudiantes de pregrado de Harvard lo estaban usando (¡sin gastar un dólar en marketing o publicidad!)

Preparar un MVP



Características de un MVP



Matriz de Priorización para el MVP

	Urgente	No urgente
Importante	1. Hacer Incluir en el MVP	2. Planear Desarrollar para un lanzamiento beta
No Importante	3. Delegar Considerar integraciones de terceras partes.	4. Eliminar Remover del Roadmap del producto

ROADMAP: la hoja de ruta del producto, primera características del primer reléase hasta la visión que tenemos del último reléase para lanzar.